DC2000 Security Target (Common Criteria)

Thales e-Security Ltd
4th/5th Floors
149 Preston Road
Brighton
BN1 6AS

Tel: +44 (0) 1273 384600
Fax: +44 (0) 1273 384601

Contents

1 Glossary

CA	Certificate Authority
CC	Common Criteria
DC2K	Datacryptor 2000
DCAP	Datacryptor Advanced Performance
DEK	Data Encryption Key
FPGA	Field-Programmable Gate Array
IP	Internet Protocol
KEK	Key Encryption Key
PCB	Printed Circuit Board
RoHS	Restriction of the use of certain hazardous substances in electrical and electronic equipment
SFP	Security Function Policy
SGSS	Secure Generic Sub-System
TOE	Target of Evaluation
TSF	TOE Security Functions
TSP	TOE Security Policy

2 References

[1] CCIMB-99-031, CCIMB-99-032, CCIMB-99-033, Common Criteria Version 2.1 Parts 1, 2 and 3
[2] 0550a109, "Key Management Specification"
[3] THALES/ENG/DEV/002 "Project Filing"
[4] 0562b245, "Datacryptor 2000 Configuration List"
[5] ENG/PM/001 "Managing Projects in the Engineering Group v1.10"
[6] THALES/ENG/DEV/006 "Problem Reporting and Change Control"
[7] THALES/ENG/DEV/012 "Document and Part Numbering v1.0"
[8] THALES/ENG/DEV/STD/006 "Software Tools Standard v1.3"
[9] THALES/OPS/005 – "In-house stores/dispatch v1.7"
[10] 1270a357, "Datacryptor 2000 Commercial Version User Manual"
[11] 1270a374, "Datacryptor AP Commercial Version User Manual"
[12] dc2000.sdt - Datacryptor 2000 SDL
[13] 0562b247, "Datacryptor 2000 Architectural Design"
[14] 0562b248, "DC2000 Representation Correspondence Analysis"
[15] 0562b243, "DC2000 Security Policy Model & Functional Specification/Security Policy Model Correspondence"
[16] Not Allocated
[17] THALES/ENG/DEV/STD/008 "Coding Standard – C v2.0"
[18] THALES/ENG/DEV/STD/011 "VHDL Coding v1.0"
[19] 0562b250, "Datacryptor 2000 Test Coverage Analysis"
[20] 0562b251, "Datacryptor 2000 Depth of Testing Analysis"
[21] 0558a363, "Datacryptor 2000 3.4a HMG System Test Specification (Commercial tests only)"
[22] 0558a368, "Datacryptor 2000 IP100 System Test Specification"
[23] 0562b252, "DC2000 Guidance Documentation Analysis"
[24] 0562b268, "DC2000 Descriptive Low-level Design"
[25] 0562b269, "DC2000 Semiformal High-level Design"
[26] 0562a218, "Datacryptor 2000 Security Target"
[27] 0562a253, "Datacryptor 2000 Vulnerabilities Analysis"
[28] 0562b254, "DC2000 Semiformal Functional Specification"
[29] 0562b276, "DC2000 Functional Testing (Common Criteria)"
[30] ENG/DEV/013 "Engineering Release Procedure"

3 Inter-Document References

The following terms defined in this Security Target may be referenced in other associated documentation. Note that the identifiers *DC2K* and *SGSS* are used to distinguish between functions provided or security objectives met by the SGSS and the DC2K.

3.1 Threats

T_extract_data_from_secure_domain
T_record_plaintext_data_from_insecure_domain
T_cryptanalyse_data_within_insecure_domain
T_access_to_secret_authentication_key
T_access_to_secret_key_exchange_alg_keys
T_access_to_keys_within_unit
T_access_to_algorithm_within_unit
T_access_to_algorithm_outside_unit
T_cryptanalyse_keys_within_insecure_domain
T_loss_of_commissioned_unit
T_tamper_with_unit
T_application_replacement
T_algorithm_replacement
T_certificate_authority_replacement
T_key_exchange_certificate_replacement

3.2 Assets

A_user_data
A_user_key
A_user_algorithm

3.3 Security Objectives

OBT_DC2K_provide_data_confidentiality
OBT_DC2K_provide_secure_key_management
OBT_DC2K_provide_secure_algorithm_load
OBT_DC2K_provide_secure_CA_load
OBT_DC2K_provide_secure_key_exchange_keyset_load

OBT_SGSS_provide_resistance_to_physical_attack
OBT_SGSS_provide_secure_application_load

OBE_protect_secure_domain
OBE_transmit_data_through_TOE
OBE_apply_suitable_TOE_mode_to_data
OBE_protect_key_material

OBE_protect_algorithms
OBE_protect_keyed_unit
OBE_check_for_unit_tamper

3.4 Security Functions

SF_DC2K_data_authentication_implementation
SF_DC2K_key_exchange_algorithm
SF_DC2K_encryption_algorithm

SF_SGSS_data_authentication_implementation
SF_SGSS_Random_Number_Generator
SF_SGSS_alarm_circuitry

4 Introduction

4.1 Security Target Identification

4.1.1 Security Target Information

Security Target Title:	Datacryptor 2000 Security Target (Common Criteria)
Part Number:	0562B218
Version Number:	001

4.1.2 Target of Evaluation Information

TOE Title:	Datacryptor 2000
Top Level Part Numbers:	Datacryptor 2000: 1600x320, Datacryptor Advanced Performance: 1600A371, 1600C371, 1600L371, 1600M371
Part Numbers:	Datacryptor 2000: 1600A321 Rev 5-8, 1600B321 Rev 006, 1600E321 Rev 7 Datacryptor Advanced Performance: 1600A372 Rev 3, 1600L372 Rev 1, 1600M372 Rev 1 Note that the issued version of a unit can be confirmed by contacting Thales e-Security.
Software Version Numbers:	Datacryptor 2000 Application Software 3.41. Datacryptor Advanced Performance Application Software 3.511.
Evaluation to include:	Communications protocols: Datacryptor 2000: Link, Frame Relay and IP5 (5Mb). Datacryptor Advanced Performance: IP10 (10Mb) and IP100 (10/100Mb).
Evaluation to exclude:	RoHS compliant version. Communications protocols: Datacryptor 2000: Link/Channelised, Link/Channelised E1 or T1, Frame Relay E1, X.25, IP – Trunk mode Datacryptor Advanced Performance: Link E3/T3,

Cryptographic Algorithms:

Key Exchange Algorithms:

Diffie-Hellman (ANSI X9.42 Hybrid1)

Data Encryption Algorithms:

Triple DES (Data Encryption Standard, as specified in FIPS PUB 46-3)

AES 128, 256 (Advanced Encryption Standard, as specified in FIPS PUB 197)

Evaluation to exclude: AES 192

Data Authentication Algorithms:

DSA (Digital Signature Algorithm, as specified in FIPS PUB 186-2)

Data Hashing Algorithms:

SHA-1 (Secure Hash Algorithm, as specified in FIPS PUB 180-2)

This Security Target has been derived using [1].

4.2 Security Target Overview

The Datacryptor 2000 (DC2K) is a range of network encryption products that support several different network protocols (e.g. IP, Frame Relay etc.). The primary purpose of the product is to provide data confidentiality. It has been designed with flexibility in mind and provides a secure soft-upgrade capability to change the network protocol and cryptographic algorithms supported. The DC2K uses public key cryptography techniques to minimise the administrative overhead of key management, and implements sophisticated measures to resist physical attack in order to safeguard key material and sensitive algorithms.

All information in the supplied deliverables that refers to 'Datacryptor 2000', 'DC2K', 'DC2000' or similar actually refers to both the Datacryptor 2000 and the Datacryptor Advanced Performance (DCAP) unless it is made clear that they refer only to one or the other either explicitly or from the context.

This document describes the security requirements and operating assumptions of the Datacryptor 2000. Section 5 gives a high level description of the physical and logical attributes of the Datacryptor 2000, and the product's scope and boundaries.

The assumed operational environment of the Datacryptor 2000 is discussed in section 6, as well as the perceived threats within that environment; a statement of the security objectives intended to counter such threats is provided in section 7.

Detailed IT security requirements are discussed in section 8 which is split into functional and assurance aspects. A Target of Evaluation Summary Specification is given in section 9, which provides a description of how the TOE IT security functions and assurance measures are met by the Datacryptor 2000.

Section 11 provides a rationale for the security target. In particular it describes the correlation between the TOE security objectives and the threats arising from the TOE's environment, a justification of the suitability of the security requirements with respect to the security objectives, and finally the means by which TOE security functions and assurance measures meet the security requirements.

4.3 Common Criteria Conformance

The Datacryptor 2000 conforms to the Common Criteria within the meaning of [1] as follows:

Part 2 Conformant
Part 3 Conformant at the Evaluation Assurance Level 4.

No claims are made with respect to the Datacryptor 2000's conformance to any Protection Profile.

5 Target Of Evaluation Description

5.1 Product Type

The Datacryptor 2000 is a range of network encryption products.

5.2 Basic Purpose

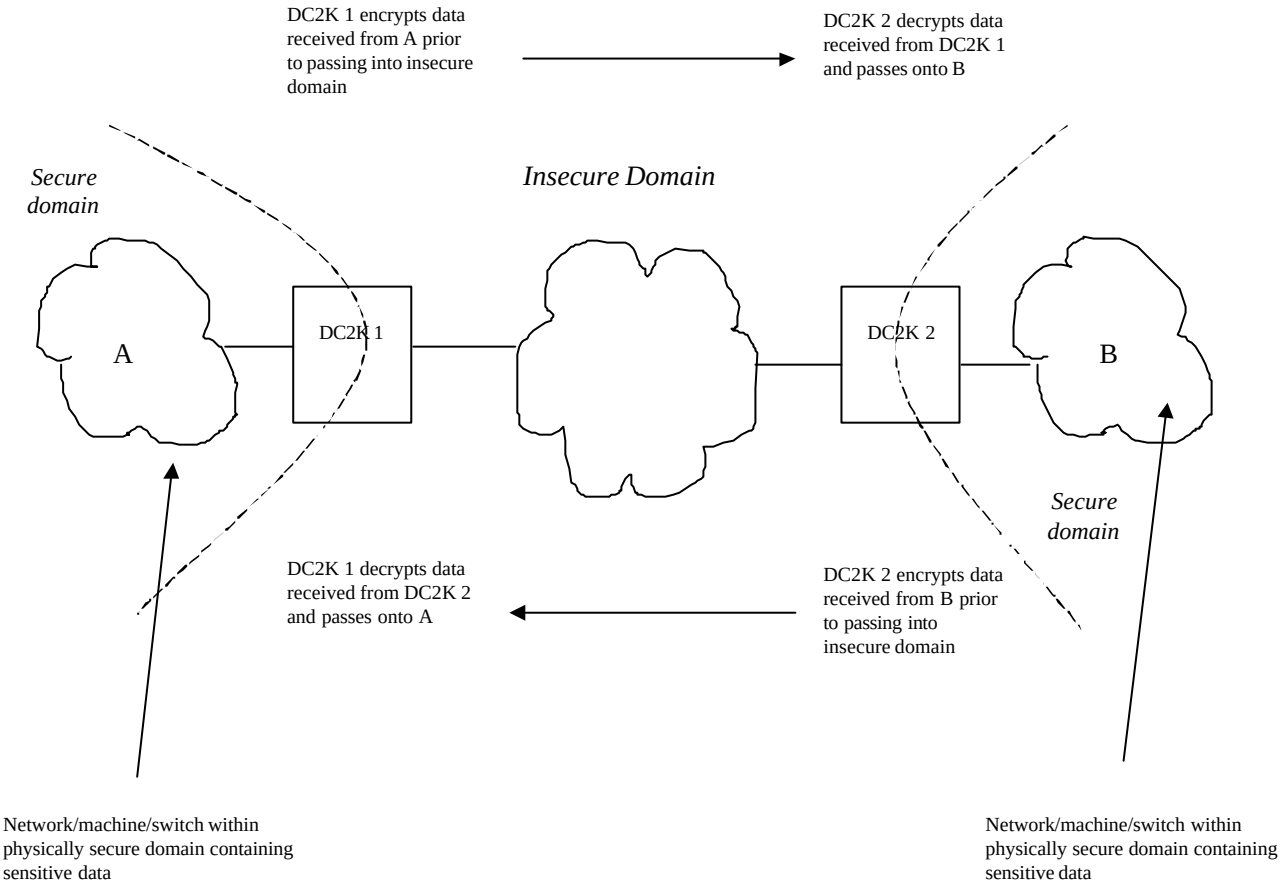


Figure 1 – DC2K usage

Figure 1 gives a pictorial representation of the primary purpose of the Datacryptor 2000. It shows a simple example of sensitive data being transmitted from one physically secured domain to another through a domain in which no physical security is assumed to be present.

At a fundamental level, the units operate in encrypting/decrypting pairs ensuring that confidentiality is afforded to data sent between the two. In this way, sensitive information may be secured whilst it is in

an insecure domain. In fact, more typically, groups of DC2Ks are deployed in various configurations to support many different network topologies and protocols, with the data passed between each pair subject to the unit's confidentiality services.

The DC2K does not provide data security services to information whilst it remains within the secure domain as defined in figure 1.

5.3 Physical Description

Several host/network interface protocols are supported (see section 4.1.2), and 2 unit management ports are provided. The unit has 5 external interfaces:

- power connector
- host data port
- network data port
- an RS232 (serial) management data port
- a 10baseT Ethernet management data port

The standard unit consists of the Secure Module, also known as the Secure Generic Sub-System (SGSS), and the DC2K baseboard.

The SGSS contains all components relevant to the secure operation of the unit. The board is subjected to physical protection using a mesh and resin technique. Alarm circuitry provided within the SGSS detects intrusion and voltage attacks, as well as movement, extremes of temperature, and pressing the erase button, where the user enables these alarms. In the event of an alarm, all the unit's sensitive contents are erased.

The baseboard provides the power interface and basic communications support.

The unit operates within the temperature range +5°C to +40°C and may be stored within the temperature range -5°C to +60°C.

5.3.1 Peripherals

As well as the unit itself, the following items are supplied to the customer.

- external power supply
- network and host cables as necessary
- CD containing unit management software and product installation and user manuals

Network and host cables are supplied where necessary to provide the physical and electrical conversion required between the Datacryptor 2000's proprietary external connectors and the communications protocol used by the customer.

The management software (a standard Windows application) runs on a PC and provides the customer with the capability to configure the unit's security and communications settings as required.

5.4 Logical Description

The Datacryptor 2000 consists of the following modules:

- The SGSS application
- The Datacryptor 2000 application
- The encryption and key exchange algorithms which are externally loaded

5.4.1 SGSS Software Application

The SGSS application runs on the SGSS. It consists of a secure bootstrap program that initialises the DC2K system and confirms the authenticity of the Datacryptor 2000 application subsequently loaded. Once the Datacryptor 2000 application has been loaded and its authenticity verified by the bootstrap, operational control passes from the SGSS application to the DC2K application.

5.4.2 Datacryptor 2000 Software Application

The DC2K application provides the following functions:

- Cryptographic verification of encryption algorithms, key exchange algorithms, Certificate Authorities and key exchange certificates subsequently loaded
- Authentication of key exchange protocol, as described in [2], section 4.2.1, (validation of public data exchange)
- Unit to management centre communication protocol and data path
- Unit to unit communication protocol and data path support
- Interface to encryption and key exchange algorithms

5.4.3 Key Exchange Algorithm

The externally loaded key exchange algorithm provides the unit with the capability to securely derive a shared key with another unit which can subsequently be used to encrypt and decrypt data transmitted between the two units (see section 5.4.4). This protocol is explained fully in [2], section 4.2.2 (steps 1 – 8) and section 4.1.1 (steps 1 – 5).

5.4.4 Encryption Algorithm

The externally loaded encryption algorithm provides the DC2K with the capability to encrypt and decrypt data transmitted and received respectively.

5.4.5 Excluded Product Functionality

Functionality excluded from the TOE includes:

- Use of the communications ports, other than in respect of the cryptographic protection given to user traffic (e.g. remote monitoring via the network port, using network management utilities via the network port)
- Hot Standby functionality.
- Remote Unit Management

6 Target of Evaluation Security Environment

6.1 Assumptions

6.1.1 Assumed Usage

The assumed usage of the Datacryptor 2000 is as shown in figure 1 of section 5.2 to provide data confidentiality to the user's information and data assets.

Additionally, it is assumed that the Datacryptor 2000's data confidentiality capability will be applied to all sensitive data to be passed between two secure domains over an insecure domain, and that appropriate policies exist with respect to:

- Choice of Key lifetime – use of keys for prolonged periods may allow cryptanalysis.
- Enabling of motion sensor as appropriate to the environment e.g. motion alarms should be enabled where theft of the unit is a threat.
- Action on pressing erase button e.g. in a hostile environment, pressing the erase button alone should have the effect of alarming the unit.
- Action in the event of suspected tampering, loss or theft of unit.

Specific advice on these aspects should be sought from the appropriate security authority.

6.1.1.1 Limitations of Use

The Datacryptor 2000 does not provide any protection to data whilst it resides in the secure environment as defined in figure 1 of section 5.2. Neither does it provide any protection where data is transmitted from the secure environment that does not pass through the unit, or where data is passed through the unit without the encryption capability of the unit having been applied by the operator.

Therefore, it is assumed that:

- The secure environment is protected to a suitable level by appropriate means
- All sensitive data is transmitted through the Datacryptor 2000
- A suitable mode of operation (e.g. encrypt mode) is applied to sensitive data passing through the Datacryptor 2000.

6.1.2 Protection of Assets

The value of the assets protected by the TOE should be appropriate for the EAL4 assurance level claimed for it: this will be relatively high, and can only be determined by the administrator in light of local conditions.

6.1.3 Assumed Environment of Operation

6.1.3.1 Management

It is assumed that, for the purposes of management, the administrator has access to appropriate management software and a PC on which to run such software.
In order to provide the necessary functionality, the management centre must be capable of encrypting and decrypting management information exchanged between itself and the unit under management.

Note that although the management centre utilises security services such as encryption, the DC2K under management will not respond to management requests unless it successfully decrypts information sent to it. In this way, the Datacryptor 2000 itself provides the security enforcing aspects necessary to ensure authorised management, since incorrect or non-existent encryption by the management centre will fail to be decrypted correctly and hence will not be acted upon. The implementation and functionality of the management centre itself, (which may be provided by the company, or developed by the user) is outside of the scope of evaluation.

6.1.3.2 Physical Protection Measures

It is assumed that physical security measures are applied to information *within the secure domain only* as appropriate to the value of the data being protected. As well as the data itself, such physical protection should be afforded to:

- Any Key material which is held externally to the unit
- Any sensitive key exchange or encryption algorithm held externally to the unit
- The unit itself while keyed
- The management centre (and any network to which the management centre is connected)

6.1.3.3 Connectivity

It is assumed that during normal operation, the unit is connected in an equivalent manner to that shown in figure 1 of section 5.2. In addition, it is assumed that when unit management is required, a separate connection is made to one of the unit's two management ports, and that neither management port is connected to the host network.

Where secret keys or sensitive algorithms are to be loaded into the unit, this must be done over a physically secured network or link.

6.1.3.4 Personnel

It is assumed that:

- Administrative personnel provided with key material enabling them to make changes to the security configuration of the unit (i.e. line mode, alarm settings, key lifetimes) are trusted appropriately.
- Access to keyed units is only provided to trusted administrative personnel
- Access to sensitive algorithms is only provided to trusted administrative personnel.

In addition it is assumed that administrative personnel have the necessary skill to operate a standard Windows application and that they have read the appropriate User Manual [10] or [11].

Note that in the context of this TOE, basic "users" of the product are those people whose data is protected by it. Since the Datacryptor 2000 is a network encryptor, the user does not have any direct interaction with the TOE – instead, administrative personnel control all product configuration and operation. As such, there are no requirements or assumptions placed on users themselves.

6.2 Threats

This section describes threats to the DC2K.

Note that some of these threats are potential attacks on the SGSS component of the DC2K. The threats have therefore been divided into two categories: threats to the SGSS and threats to the DC2K. (However threats to the SGSS, when the SGSS is installed as a component of the DC2K, are also threats to the DC2K itself.)

6.2.1 Extraction of Data from Within the Secure Domain

Inter-document reference T_extract_data_from_secure_domain

This is a threat to the DC2K.

The DC2K does not provide any security services to data whilst it resides within the secure domain. Instead it operates by encrypting any data passed into its host port from the host network, machine or switch within the secure domain, prior to passing the encrypted data out into the insecure domain.

A threat exists whereby a threat agent could gain access to unencrypted data whilst it resides within this area.

6.2.1.1 Attack

The attack is to gain physical access to the secure domain and record data from within that domain.

6.2.1.2 Asset

Inter-document Reference – A_user_data

The asset under threat is the user's data.

6.2.1.3 Threat Agent

Expertise:	A low level of expertise is required. Data within this domain is not subject to any technical protection, but the threat agent will require the capability to extract the data and decode it from standard network protocols. Such decoding is within

	the capability of any individual with a basic understanding of common, public domain network protocols.
Resource:	Limited resource is required. Some means of recording and decoding the data from the secure domain is required. However, such tools are available, and, depending on the protocol, are relatively cheap. IP recording and decoding tools for example are primarily simple software Windows applications.
Motivation:	Since the value of the asset is relatively high, it must be assumed that motivation to mount this attack is high.
Vulnerabilities Exploited:	Within the definition of the threat, the asset is not subject to the protection of the TOE. Therefore no TOE vulnerabilities need be exploited in order to mount a successful attack.
Opportunity:	If the threat agent has obtained access to the secure domain (either by virtue of authorisation, or by breaching physical security measures), opportunity to mount the attack is high. Otherwise, opportunity is extremely limited.

6.2.2 Recording of Plaintext Data Leaked into Insecure Domain

Inter-document reference T_record_plaintext_data_from_insecure_domain

This is a threat to the DC2K.

Unencrypted data may be present in the insecure domain for two reasons:

- The data has not been sent via a channel that passes through the DC2K prior to entering the insecure domain.
- The data has been sent through a channel that passes through the DC2K, but the DC2K's encryption capability has not been applied to the data. i.e. the unit's line mode has been set to an insecure mode.

A threat exists whereby a threat agent records such unencrypted data from the insecure domain.

6.2.2.1 Attack

The attack is to record any unencrypted data from the insecure domain.

6.2.2.2 Asset

Inter-document Reference – A_user_data

The asset under threat is the user's data.

6.2.2.3 Threat Agent

Expertise:	A low level of expertise is required. Unencrypted data within this domain is not subject to any technical protection, but the threat agent will require the capability to extract the data and decode it from standard network protocols. Such decoding is within the capability of any individual with a basic understanding of common, public domain network protocols.
Resource:	Limited resource is required. Some means of recording and decoding the data from the insecure domain is required. However, such tools are available, and, depending on the protocol, are relatively cheap. IP recording and decoding tools for example are primarily simple software Windows applications.
Motivation:	Since the value of the asset is relatively high, it must be assumed that motivation to mount this attack is high.
Vulnerabilities Exploited:	Within the definition of the threat, the asset has not been subject to the protection of the TOE. Therefore no TOE vulnerabilities need be exploited in order to mount a successful attack.
Opportunity:	Where such unencrypted data is present in the insecure domain, the opportunity for the threat agent to mount this attack is high.

6.2.3 Cryptanalysis of data in the insecure domain

Inter-document Reference – T_cryptanalyse_data_within_insecure_domain

This is a threat to the DC2K.

In normal usage, it is anticipated that sensitive data residing within the secure domain will be subjected to the TOE's data confidentiality measures prior to it being passed into an insecure domain.

A threat exists for an attacker to record encrypted data sent across the insecure domain and subject it to cryptanalysis in an attempt to discover the underlying plaintext data.

6.2.3.1 Attack

The attack is to record encrypted data from the insecure domain, decode it and perform cryptanalysis to reveal the underlying plaintext data.

6.2.3.2 Asset

Inter-document Reference – A_user_data

The asset under threat is the user's data.

6.2.3.3 Threat Agent

Expertise:	Assuming that an appropriate encryption algorithm is used, its implementation is not flawed, and that key material has not been leaked, a high level of expertise is required to successfully gain plaintext from encrypted data.
Resource:	The resource requirements to mount an attack of this type are high – a very large amount of computing power, either distributed or within one unit would be required.
Motivation:	Since the value of the asset is relatively high, it must be assumed that motivation to mount this attack is high.
Vulnerabilities Exploited:	If a vulnerability were present in the TOE's encryption algorithm or in its implementation, this may be exploited to decrease the level of expertise or resource required for success.
Opportunity:	Where such encrypted data is present in the insecure domain, the opportunity for the threat agent to mount this attack is high.

6.2.4 Exposure of Secret Authentication Key

Inter-document Reference – T_access_to_secret_authentication_key

This is a threat to the DC2K.

Secret authentication keys (although not loaded into the unit) are used to generate signed key exchange certificates.

If it were possible for an attacker to gain access to such material as it exists externally to the unit, it may be possible for him to ultimately determine the key used to encrypt the user's data, and then use the key to decrypt encrypted data.

There are two means by which access may be provided to a threat agent:

- An individual with authorised access to the material may leak the data intentionally or unintentionally
- An unauthorised individual may breach physical measures to gain access to the material

6.2.4.1 Attack

A threat agent who has gained access to the secret authentication key may forge signed key exchange certificates. This would allow an active "man-in-the-middle" attack to be mounted between two units whereby both units are spoofed into establishing a shared key encryption key with the threat agent rather than the other unit. A similarly shared data encryption key could then be generated using the rogue key encryption key, and then used to decrypt the user's data.

6.2.4.2 Asset

Inter-document Reference – A_user_key
A_user_data

The asset under threat is the user group secret authentication key, exposure of which may ultimately lead to exposure of the user's data.

6.2.4.3 Threat Agent

Expertise:	A high level of expertise is required. Even assuming successful access to the secret authentication key, the attack is a very sophisticated real-time active attack. It requires insertion and deletion of data from the line between two units without either unit "noticing" the presence of the third party.
Resource:	A high level of resource is required. Equipment to insert and remove traffic from the line in real-time is required, as is equipment which can spoof the entire key exchange protocol, and subsequently react appropriately to any peer-unit requests i.e. subsequent data encryption key updates in a timely fashion.
Motivation:	Since the value of the asset is relatively high, it must be assumed that motivation to mount this attack is high. In particular, repeated application of this attack could give rise to the successful decryption of traffic within the entire lifetime of the secret authentication key.
Vulnerabilities Exploited:	Within the definition of the threat, the asset has not been subject to the protection of the TOE. Therefore no TOE vulnerabilities need be exploited in order to mount a successful attack.

Opportunity: If the threat agent has obtained access to the secure domain (either by virtue of authorisation, or by breaching physical security measures), opportunity to mount the attack is high.

Otherwise, opportunity is extremely limited.

6.2.5 Exposure of Secret Keys Used in the Key Exchange Algorithm

Inter-document Reference – T_access_to_secret_key_exchange_alg_keys

This is a threat to the DC2K.

In some modes of use, secret key exchange algorithm keys are loaded into the unit from an external source.

If it were possible for an attacker to gain access to such material as it exists externally to the unit, it may be possible for him to ultimately determine the key used to encrypt the user's data, and then use the key to decrypt encrypted data.

There are three means by which access may be provided to a threat agent:

- An individual with authorised access to the material may leak the data intentionally or unintentionally
- An unauthorised individual may breach physical measures to gain access to the material
- The unit may be commissioned with secret key material over an unprotected link or network, to which a threat agent may have access.

6.2.5.1 Attack

Long term secret keys are input by both units participating in the KEK derivation algorithm, and in some modes of operation, these are loaded from an external source. However, the algorithm also utilises relatively substantial quantities of one-time random data generated by the unit's themselves. This random data is never exposed outside the unit. Assuming that the units' random number generator is operating properly, a threat agent would have to guess (or exhaust over) these random values to be able to determine the key encryption key and subsequently the data encryption key used.

6.2.5.2 Asset

Inter-document Reference – A_user_key
A_user_data

The asset under threat is the user's secret key exchange algorithm keys, exposure of which may subsequently lead to exposure of the user's data.

6.2.5.3 Threat Agent

Expertise:	A high level of expertise is required. The threat agent would have to efficiently exhaust over all possible values of one-time random input to the key exchange algorithm to be able to determine the key encryption key established between the two units, and subsequently determine the data encryption key.
Resource:	An extremely high level of resource is required. Assuming that at least one of the unit's random number generators is operating correctly, a huge amount of computing resource would be required to exhaust over all possible one-time random inputs to the algorithm.
Motivation:	Since the value of the asset is relatively high, it must be assumed that motivation to mount this attack is high. In particular, repeated application of this attack could give rise to the successful decryption of traffic within the entire lifetime of the secret key exchange algorithm key.
Vulnerabilities Exploited:	Within the definition of the threat, the asset has not been subject to the protection of the TOE. Therefore no TOE vulnerabilities need be exploited in order to mount a successful attack.
Opportunity:	If the threat agent has obtained access to the secure domain (either by virtue of authorisation, or by breaching physical security measures), or to an insecure commissioning session, opportunity to mount the attack is high. Otherwise, opportunity is extremely limited.

6.2.6 Discovery or Substitution of any Key Material Stored within Unit

Inter-document Reference – T_access_to_keys_within_unit

This is a threat to the SGSS (and hence also to the DC2K – see 6.2).

Most secret key material (i.e. key encryption keys and data encryption keys) is generated and stored internally by the DC2K unit. If it were possible for a threat agent to discover such key values or substitute them for values known to him, it may be possible for that information to be used in the decryption of user data.

6.2.6.1 Attack

The attack is to gain access to the unit's sensitive storage areas and extract some or all of their contents without triggering an alarm (which would cause the sensitive contents to be erased).

6.2.6.2 Asset

Inter-document Reference – A_user_key
A_user_data

The asset under threat is potentially all of the user's secret key material, exposure of which may subsequently lead to exposure of the user's data.

6.2.6.3 Threat Agent

Expertise:	A high level of expertise is required. The threat agent would have to extract keys from the unit without triggering an alarm.
Resource:	A high level of resource is required. The threat agent would require sophisticated and specialised equipment to breach the unit's physical protection mechanisms. Such equipment might include X-ray capability, extremely fine drills, chemical solvents etc.
Motivation:	Since the value of the asset is relatively high, it must be assumed that motivation to mount this attack is high. In particular, information gained from this attack could be used to determine keys and hence traffic for the lifetime of the key or keys extracted.
Vulnerabilities Exploited:	If a vulnerability were present in the TOE's physical protection design or implementation, this may be exploited to decrease the level of expertise or resource required for success.
Opportunity:	Opportunity to undertake this attack is limited by the availability of access to the unit itself.

6.2.7 Extraction of Sensitive Cryptographic Algorithm From Unit

Inter-document Reference – T_access_to_algorithm_within_unit

This is a threat to the SGSS (and hence also to the DC2K – see 6.2).

In some cases, the cryptographic algorithms used to provide data and key confidentiality services to the user are sensitive. Extraction of such a sensitive cryptographic algorithm from the unit may be undesirable for political reasons, and possibly assists future cryptanalysis.

6.2.7.1 Attack

The attack is to gain access to the unit's sensitive storage areas and extract some or all of their contents without triggering an alarm (which would cause the sensitive contents to be erased).

6.2.7.2 Asset

Inter-document Reference – A_user_algorithm

The asset under threat is the user's sensitive cryptographic algorithms.

6.2.7.3 Threat Agent

Expertise:	A high level of expertise is required. The threat agent would have to extract the algorithm from the unit without triggering an alarm.
Resource:	A high level of resource is required. The threat agent would require sophisticated and specialised equipment to breach the unit's physical protection mechanisms. Such equipment might include X-ray capability, extremely fine drills, chemical solvents etc.
Motivation:	Since the value of the asset is high, it must be assumed that motivation to mount this attack is high. In particular, information gained from this attack could be used to determine information about protective measures applied by other secure applications owned by the user.
Vulnerabilities Exploited:	If a vulnerability were present in the TOE's physical protection design or implementation, this may be exploited to decrease the level of expertise or resource required for success.
Opportunity:	Opportunity to undertake this attack is limited by the availability of access to the unit itself.

6.2.8 Compromise of Sensitive Cryptographic Algorithm when external to unit

Inter-document Reference – T_access_to_algorithm_outside_unit

This is a threat to the DC2K.

In some cases, the cryptographic algorithms used to provide data and key confidentiality services to the user are sensitive. Exposure of such a sensitive cryptographic algorithm when it is stored externally to the unit may be undesirable for political reasons, and possibly assists future cryptanalysis.

6.2.8.1 Asset

Inter-document Reference – A_user_algorithm

The asset under threat is the user's sensitive cryptographic algorithms.

6.2.8.2 Attack

The attack is to gain access to the user's sensitive cryptographic algorithms, wherever those may be stored, with intent to gain expertise in cryptanalysis of the user's communications.

6.2.8.3 Threat Agent

Expertise:	A low level of expertise is required. When stored externally to the unit, the algorithm is not subject to any technical protection
Resource:	Limited resource is required.
Motivation:	Since the value of the asset is high, it must be assumed that motivation to mount this attack is high. In particular, information gained from this attack could be used to determine information about protective measures applied by other secure applications owned by the user.
Vulnerabilities Exploited:	Within the definition of the threat, the asset has not been subject to the protection of the TOE. Therefore no TOE vulnerabilities need be exploited in order to mount a successful attack.
Opportunity:	If the threat agent has obtained access to the secure domain (either by virtue of authorisation, or by breaching physical security measures), opportunity to mount the attack is high. Otherwise, opportunity is extremely limited.

6.2.9 Cryptanalysis of Encrypted Keys in the insecure domain

Inter-document Reference – T_cryptanalyse_keys_within_insecure_domain

This is a threat to the DC2K.

Two communicating Datacryptor 2000s must undertake a key exchange protocol prior to exchanging encrypted data. Full details of the key exchange protocols are provided in sections 4.2.2 (steps 1 – 8) and 4.1.1 (steps 1 – 5) of [2]; the use of public key cryptography allows two commissioned units operating within the same user group to negotiate a shared Key Encryption Key, and subsequently a

data encryption key. These protocols operate in such a way that there is no requirement for any secret data to be transmitted from either unit.

A threat exists whereby an attacker may perform cryptanalysis on the key exchange protocols to determine the key encryption keys and or data encryption keys subsequently used by the unit. An alternative attack may be to force the re-use of a key, possibly leading to easier cryptanalysis of encrypted data.

6.2.9.1 Attack

The attack is to perform cryptanalysis on the DC2K's key exchange protocols with intent to use key information to decrypt traffic transmitted between the units.

6.2.9.2 Asset

Inter-document Reference – A_user_key
A_user_data

The asset under threat is the user's key encryption keys and data encryption keys.

6.2.9.3 Threat Agent

Expertise:	Assuming that an appropriate key exchange and key encryption algorithms are used, and their implementations are not flawed, a high level of expertise is required to successfully gain keys from the key exchange protocols.
Resource:	The resource requirements to mount an attack of this type are high – a very large amount of computing power, either distributed or within one unit would be required.
Motivation:	Since the value of the asset is relatively high, it must be assumed that motivation to mount this attack is high. Access to the unit's keys could potentially lead plaintext for the lifetime of the key exposed.
Vulnerabilities Exploited:	If a vulnerability were present in the TOE's key exchange or key encryption algorithms or implementations, this may be exploited to decrease the level of expertise or resource required for success.
Opportunity:	Where such encrypted keys are present in the insecure domain, the opportunity for the threat agent to mount this attack is high.

6.2.10 Unit theft or loss

Inter-document Reference – T_loss_of_commissioned_unit

This is a threat to the DC2K.

A commissioned unit that is subsequently lost or stolen has all the necessary keys in place to engage in an encrypted session with another unit, that may still be encrypting legitimate user data. If the unit's disappearance goes unnoticed for a period of time, the user may unknowingly be sending his information to an attacker.

6.2.10.1 Attack

The attack consists of a threat agent gaining access to a commissioned unit and using the unit to decrypt traffic sent to it by a unit still within the user's possession.

6.2.10.2 Asset

Inter-document Reference – A_user_data

The asset under threat is the user's data.

6.2.10.3 Threat Agent

Expertise:	A low level of expertise is required to steal the box, and to subsequently use it to decrypt traffic transmitted by another unit within the same user group.
Resource:	The resource requirement to mount an attack of this type is low. No specialist equipment is required.
Motivation:	Since the value of the asset is relatively high, it must be assumed that motivation to mount this attack is high.
Vulnerabilities Exploited:	Within the definition of the threat, the asset has not been subject to the protection of the TOE. Therefore no TOE vulnerabilities need be exploited in order to mount a successful attack.
Opportunity:	If the threat agent has obtained access to the secure domain (either by virtue of authorisation, or by breaching physical security measures), or a user has simply lost a unit within the insecure domain, opportunity to mount the attack is high. Otherwise, opportunity is extremely limited.

6.2.11 Unit tampering

Inter-document Reference – T_tamper_with_unit

This is a threat to the DC2K.

It may be possible to tamper with a unit that is transmitting encrypted data in such a way that the security provided by the unit is undermined. If such tampering were to go unnoticed, a large amount of data could be leaked.

6.2.11.1 Attack

The attack consists of a threat agent gaining access to a unit and tampering with it so as to cause plaintext data to be leaked into the insecure domain. Such unprotected data could then be recorded from within the insecure domain.

6.2.11.2 Asset

Inter-document Reference – A_user_data

The asset under threat is the user's data.

6.2.11.3 Threat Agent

Expertise:	A moderate level of expertise is required to alter the box in such a way that it causes insecure operation and the tampering goes unnoticed by the user.
Resource:	The resource requirement for such an attack is moderate – specialist equipment may be required to alter the box in an unnoticeable fashion.
Motivation:	Since the value of the asset is relatively high, it must be assumed that motivation to mount this attack is high.
Vulnerabilities Exploited:	Within the definition of the threat, the asset has not been subject to the protection of the TOE. Therefore no TOE vulnerabilities need be exploited in order to mount a successful attack.
Opportunity:	If the threat agent has obtained access to the secure domain (either by virtue of authorisation, or by breaching physical security measures), opportunity to mount the attack is high. Otherwise, opportunity is extremely limited.

6.2.12 Loading of Malicious Application Code

Inter-document Reference T_application_replacement

This is a threat to the DC2K.

Amongst other security critical functions, the Datacryptor 2000's application code controls the cryptographic protection measures that are provided to the user's data. If that application were

replaced by a rogue application that subverted the security provided by the application, it is possible that data, keys and algorithms could all be exposed.

6.2.12.1 Asset

Inter-document Reference: *A_user_data*
A_user_key
A_user_algorithm

The assets at threat from this attack are the user's data, keys and algorithms

6.2.12.2 Attack

The attack requires the generation of a substitute application that induces insecurity into the system. In addition, the application must be formatted and digitally signed by the secret authentication key corresponding to the public key embedded in the SGSS's secure bootstrap code. Having generated such an application, the threat agent also needs to load it into the unit.

6.2.12.3 Threat Agent

Expertise:	Assuming that the data authentication algorithm and its implementations are not flawed, and that the code secret authentication key is unavailable, a high level of expertise is required to successfully generate an application that will be verified by the SGSS's bootstrap.
Resource:	The resource requirements to mount an attack of this type are extremely high – a very large amount of computing power, either distributed or within one unit would be required to generate an application whose authenticity would be verified by the SGSS. Furthermore, the only way to achieve the attack is by iteratively generating *and attempting to load* the application into a unit.
Motivation:	Since the value of the asset is relatively high, it must be assumed that motivation to mount this attack is high. A sufficiently insecure application might yield plaintext, keys and algorithms to the threat agent.
Vulnerabilities Exploited:	If a vulnerability were present in the SGSS's secure bootstrap implementation, or in the data authentication algorithm used, this may be exploited to decrease the level of expertise or resource required for success.
Opportunity:	Where an attacker has access to the unit into which to load a rogue application (either by virtue of being provided with authorised access,

or by breaching physical security measures), the opportunity for the threat agent to mount this attack is present.

Otherwise, opportunity is extremely limited.

6.2.13 Loading of Malicious Encryption or Key Exchange Algorithm

Inter-document Reference T_algorithm_replacement

This is a threat to the DC2K.

The Datacryptor 2000's encryption and key exchange algorithms are soft-loaded under the cryptographic control of the application. If those algorithms were replaced by rogue algorithms that performed poor (or non-existent) data or key encryption, the user's keys and data may be exposed.

6.2.13.1 Asset

Inter-document Reference: *A_user_data*
A_user_key

The assets at threat from this attack are the user's data and keys.

6.2.13.2 Attack

The attack requires the generation of substitute algorithms that induces insecurity into the system. In addition, the algorithms must be formatted and digitally signed by the secret authentication key corresponding to the public key embedded in the Datacryptor's application code. Having generated such algorithms, the threat agent also needs to be able to load them into the unit.

6.2.13.3 Threat Agent

Expertise: Assuming that the data authentication algorithm and its implementations are not flawed, and that the secret algorithm authentication key is unavailable, a high level of expertise is required to successfully generate an algorithm that will be verified by the DC2K application.

Resource: The resource requirements to mount an attack of this type are extremely high – a very large amount of computing power, either distributed or within one unit would be required to generate an algorithm whose authenticity would be verified by the DC2K application. Furthermore, the only way to achieve the attack is by iteratively generating *and attempting to load* the algorithms into a unit.

Motivation:	Since the value of the asset is relatively high, it must be assumed that motivation to mount this attack is high. A sufficiently insecure algorithm might yield plaintext and keys to the threat agent.
Vulnerabilities Exploited:	If a vulnerability were present in the DC2K's algorithm authentication implementation, or in the data authentication algorithm used, this may be exploited to decrease the level of expertise or resource required for success.
Opportunity:	Where an attacker has access to the unit into which to load a rogue algorithm (either by virtue of being provided with authorised access, or by breaching physical security measures), the opportunity for the threat agent to mount this attack is present. Otherwise, opportunity is extremely limited.

6.2.14 Loading of Certificate Authorities Known to the Attacker

Inter-document Reference T_certificate_authority_replacement

This is a threat to the DC2K.

The Datacryptor 2000's Certificate Authorities are signed and loaded under the cryptographic control of the application. If this key material were replaced with equivalent key material known to the threat agent *in two units*, it may be possible for him subsequently to load consistent known or degenerate key exchange keysets into both units. Ultimately this attack might lead to the threat agent being able to determine the key used to encrypt the user's data, and use this key to decrypt encrypted data.

6.2.14.1 Attack

Long term public and secret keys are input by both units participating in the KEK derivation algorithm. However, the algorithm also utilises relatively substantial quantities of one-time random data generated by the unit's themselves. This random data is never exposed outside the unit. Assuming that the units' random number generator is operating properly, a threat agent would have to guess (or exhaust over) these random values to be able to determine the key encryption key and subsequently the data encryption key used.

6.2.14.2 Asset

Inter-document Reference – A_user_key

The asset directly under threat is the integrity of the user's Certificate Authority, alteration of which may subsequently lead to the exposure of key encryption keys, data encryption keys, and finally exposure of the user's data.

6.2.14.3 Threat Agent

Expertise:	A high level of expertise is required. Firstly, the threat agent would have to generate a signed Certificate Authority whose authenticity would be verified by the DC2K application, and load it into the unit, together with known key exchange keysets authorised by that CA. Secondly, even having achieved this, he would have to efficiently exhaust over all possible values of one-time random input to the key exchange algorithm to be able to determine the key encryption key established between the two units, and subsequently determine the data encryption key.
Resource:	An extremely high level of resource is required both to generate the signed CA and to determine the one-time random input to the key encryption key generation process.
Motivation:	Since the value of the asset is relatively high, it must be assumed that motivation to mount this attack is also high. In particular, repeated application of this attack could give rise to the successful decryption of traffic within the entire lifetime of the CA.
Vulnerabilities Exploited:	If a vulnerability were present in the DC2K's certificate authentication implementation, or in the data authentication algorithm used, this may be exploited to decrease the level of expertise or resource required for success.
Opportunity:	Where an attacker has access to the unit into which to load a replacement CA and keyset (either by virtue of being provided with authorised access, or by breaching physical security measures), the opportunity for the threat agent to mount this attack is present. Otherwise, opportunity is extremely limited.

6.2.15 Loading of Key Exchange Certificates Known to the Attacker

Inter-document Reference T_key_exchange_certificate_replacement

This is a threat to the DC2K.

The Datacryptor 2000's Key Exchange public key certificates (and in some instances the corresponding secret key) are signed and loaded under the cryptographic control of the application. If this keyset were replaced with equivalent key material known to the threat agent, it may be possible for him to determine the key encryption key and data encryption key, and finally use this key to decrypt encrypted data.

6.2.15.1 Attack

Long term public and secret keys are input by both units participating in the KEK derivation algorithm. However, the algorithm also utilises relatively substantial quantities of one-time random data generated by the unit's themselves. This random data is never exposed outside the unit. Assuming that the units' random number generator is operating properly, a threat agent would have to guess (or exhaust over) these random values to be able to determine the key encryption key and subsequently the data encryption key used.

6.2.15.2 Asset

Inter-document Reference – A_user_key

The asset directly under threat is the user's key exchange algorithm keys, exposure of which may subsequently lead to key encryption keys, data encryption keys, and finally exposure of the user's data.

6.2.15.3 Threat Agent

Expertise:	A high level of expertise is required. Firstly, the threat agent would have to generate a signed key exchange keyset whose authenticity would be verified by the DC2K application, and load it into the unit. Secondly, even having achieved this, he would have to efficiently exhaust over all possible values of one-time random input to the key exchange algorithm to be able to determine the key encryption key established between the two units, and subsequently determine the data encryption key.
Resource:	An extremely high level of resource is required both to generate the signed keyset and to determine the one-time random input to the key encryption key generation process.
Motivation:	Since the value of the asset is relatively high, it must be assumed that motivation to mount this attack is also high. In particular, repeated application of this attack could give rise to the successful decryption of traffic within the entire lifetime of the key exchange certificates.
Vulnerabilities Exploited:	If a vulnerability were present in the DC2K's key exchange keyset authentication implementation, or in the data authentication algorithm used itself, this may be exploited to decrease the level of expertise or resource required for success.
Opportunity:	Where an attacker has access to the unit into which to load a replacement keyset (either by virtue of being provided with authorised access, or by breaching physical security measures), the opportunity for the threat agent to mount this attack is present.

Otherwise, opportunity is extremely limited.

6.3 Organisational Security Policies

No claims are made regarding the Datacryptor 2000's compliance with specific organisational security policies.

7 Security Objectives

7.1 Security Objectives for the Target of Evaluation

7.1.1 Datacryptor Security Objectives

7.1.1.1 Provision of Data Confidentiality Service

Inter-document Reference OBT_DC2K_provide_data_confidentiality

The DC2K shall provide the option of a confidentiality service to all data that is transmitted through it.

7.1.1.2 Provision of Secure Key Management Service

Inter-document Reference OBT_DC2K_provide_secure_key_management

The DC2K shall provide a means of securely exchanging key material for use in the provision of data confidentiality.

7.1.1.3 Provision of Secure Algorithm Loading Capability

Inter-document Reference OBT_DC2K_provide_secure_algorithm_load

The DC2K shall provide a means by which the authenticity of a cryptographic algorithm may be itself cryptographically verified prior to its loading and usage.

7.1.1.4 Provision of Secure Certificate Authority Loading Capability

Inter-document Reference OBT_DC2K_provide_secure_CA_load

The DC2K shall provide a means by which the authenticity of a Certificate Authority may be cryptographically verified prior to its loading and usage.

7.1.1.5 Provision of Secure Key Exchange Keyset Loading Capability

Inter-document Reference OBT_DC2K_provide_secure_key_exchange_keyset_load

The DC2K shall provide a means by which the authenticity of a Key Exchange Keyset may be cryptographically verified prior to its loading and usage.

7.1.2 SGSS Security Objectives

7.1.2.1 Provision of Physical Security Measures to Sensitive Data Stored Within TOE

Inter-document Reference OBT_SGSS_provide_resistance_to_physical_attack

The SGSS shall provide physical resistance to direct technical attack aimed at the extraction of sensitive data from within the unit.

7.1.2.2 Provision of Secure Application Loading Capability

Inter-document Reference OBT_SGSS_provide_secure_application_load

The SGSS shall provide a means by which the authenticity of a Datacryptor 2000 application may be cryptographically verified prior to its loading and storage.

7.2 Security Objectives for the Environment

7.2.1 Application of Physical Protection to the Secure Domain

Inter-document Reference - OBE_protect_secure_domain

Physical protection measures i.e. securely locked premises, guards etc. must be applied as necessary to the secure domain in which sensitive and otherwise unprotected data resides. The value of the assets protected by the TOE should be appropriate for the EAL4 assurance level claimed for it: this will be relatively high, and can only be determined by the administrator in light of local conditions.

7.2.2 Application of TOE to all sensitive data transmitted

Inter-document Reference - OBE_transmit_data_through_TOE

All sensitive data held within the secure domain must be passed through the TOE prior to it reaching the insecure domain.

7.2.3 Appropriate Use of TOE's Protection Capabilities

Inter-document Reference - OBE_apply_suitable_TOE_mode_to_data

The DC2K's "encrypt line mode" must be applied to all sensitive data passing through the unit.

7.2.4 Application of Physical Security to External Key Material

Inter-document Reference - OBE_protect_key_material

Physical protection measures i.e. securely locked premises, guards etc. must be applied as necessary to sensitive key material where this is stored externally to the unit.

7.2.5 Application of Physical Security to External Sensitive Algorithms

Inter-document Reference - OBE_protect_algorithms

Physical protection measures i.e. securely locked premises, guards etc. must be applied as necessary to sensitive algorithms where these are stored externally to the unit.

7.2.6 Application of Physical Security to Unit When Keyed

Inter-document Reference - OBE_protect_keyed_unit

Physical protection measures i.e. securely locked premises, guards etc. must be applied as necessary to units that have been commissioned.

7.2.7 Check For Signs of Unit Tampering

Inter-document Reference - OBE_check_for_unit_tamper

Units should be checked periodically for signs of tampering. If tampering is deemed to have taken place, this should be reported immediately to the appropriate authority.

8 IT Security Requirements

8.1 Target of Evaluation Security Requirements

8.1.1 Target of Evaluation Security Functional Requirements

The following security functional requirements, defined in the form of components extracted from [1], are required to fully support the TOE security objectives. The assignment operation on the security requirements is indicated by normal underlined text in square brackets.

Section 11.4.1 lists the TOE security functions that meet each of the security functional requirements. Please see section 9.1 for a discussion of the TOE security functions provided by the SGSS and DC2K.

(Note that although assurance component AVA_SOF.1 is included in the TOE Security Assurance requirements, all TOE Security Functions realised by a probabilistic or permutational mechanism are cryptographic. Hence a statement regarding their strength level is outside the scope of this Security Target.)

8.1.1.1 FCS_CKM.1 – Cryptographic Key Generation

FCS_CKM.1.1
The TSF shall generate cryptographic keys in accordance with specified cryptographic key generation algorithms [described in [2]] and specified cryptographic key sizes [as specified in section 4.1.2] that meet the following: [algorithm specification defined or standards referenced in section 4.1.2].

8.1.1.2 FCS_CKM.2 – Cryptographic Key Distribution

FCS_CKM.2.1
The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [as defined in [2]] that meets the following: [standards referenced in section 4.1.2].

8.1.1.3 FCS_COP.1 – Cryptographic Operation

FCS_COP.1.1
The TSF shall perform [data authentication[1], key exchange protocol, data encryption] in accordance with a specified cryptographic algorithm [listed in section 4.1.2] and cryptographic key sizes [as in the algorithm specification] that meet the following: [standards referenced within section 4.1.2].

[1] In the context of a cryptographic product, data authentication has a precise meaning; it is a means by which the receiver of data can cryptographically ascertain its origin, such that the sender of the data cannot masquerade as someone else.

8.1.1.4 FPT_PHP.3 – Resistance to Physical Attack

FPT_PHP.3.1
The TSF shall resist [physical intrusion, high and low voltage attacks and attacks requiring temperature extremes] to the [SGSS component of the TOE] by responding automatically such that the TSP is not violated.

8.1.2 Target of Evaluation Security Assurance Requirements

The TOE is compliant with the assurance components required by Evaluation Assurance Level 4 (see [1]).

8.2 Security Requirements for the IT Environment

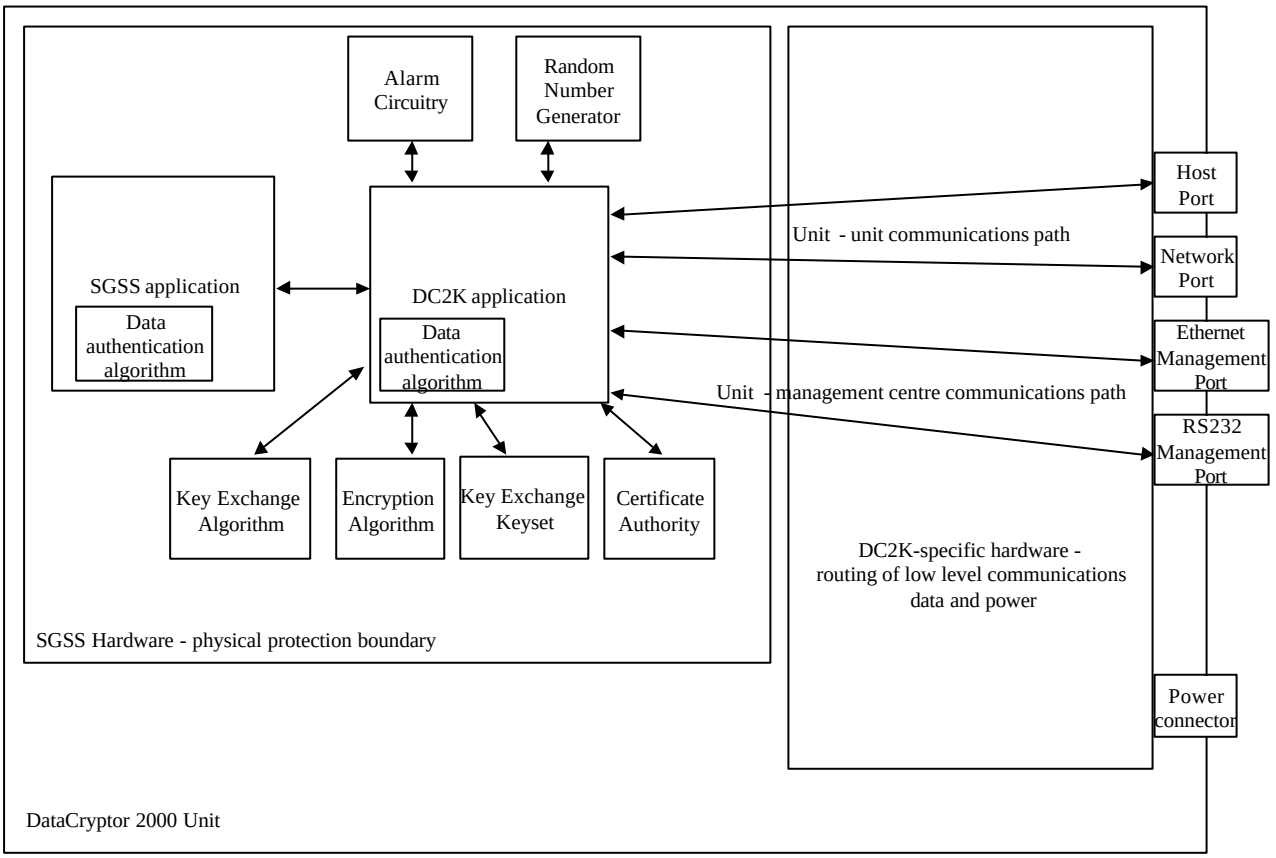
There are no security requirements for the Datacryptor 2000's assumed IT environment. (See sections 6.1.3.1 and 9.1.7).

9 Target of Evaluation Summary Specification

9.1 Target of Evaluation Security Functions

For reference a previous Security Target [26] is supplied that covered the same Security Functions and obtained EAL4 against the UK Common Criteria.

(Note that although assurance component AVA_SOF.1 is included in the TOE Security Assurance requirements, all TOE Security Functions realised by a probabilistic or permutational mechanism are cryptographic. Hence a statement regarding their strength level is outside the scope of this Security Target.)



9.1.1 TOE System Architecture

Figure 2 – Datacryptor 2000 Architecture

Figure 2 depicts the high level architecture of the Datacryptor 2000. It demonstrates the organisation of the product at a logical and physical level and provides a context in which to discuss the instantiation of the TOE's security functions.

9.1.2 SGSS Application

The SGSS application performs 2 functions:

- Non security-relevant system start-up – this aspect will not be discussed here
- Cryptographic authentication of any DC2K application to be loaded using a data authentication algorithm listed in section 4.1.2. This aspect is discussed below.

The SGSS contains an implementation of a data authentication algorithm, as listed in section 4.1.2. In addition, at manufacture time, a public key value is embedded in the SGSS application. When the DC2K application is generated, a digital signature is generated over the application using the corresponding secret key, which is held securely at the development site. The application is concatenated with the signature.

On loading the DC2K application, the signature concatenated with the application is verified by the SGSS application's data authentication implementation using the public key embedded within it. If the verification is successful, the application is loaded into the SGSS hardware and may be used. Otherwise, the application is rejected and cannot be loaded into the unit.

In this way, only DC2K applications that have been signed by the manufacturer may be run in the unit.

Inter-document Reference SF_SGSS_data_authentication_implementation

9.1.3 SGSS Hardware

9.1.3.1 Random Number Generator

The SGSS hardware contains a hardware random number generator that generates high quality random numbers for use in the Key Exchange Protocol (see sections 4.2.2 (steps 1 – 8) and 4.1.1 (steps 1 – 5) of [2]). The random number generator output is subject to frequent background diagnostic statistical testing, failure of which causes the unit to cease transmission of data. This ensures that all random numbers used for security relevant purposes are of high quality.

Inter-document Reference SF_SGSS_Random_Number_Generator

9.1.3.2 Alarm Circuitry

The physical security provided by the SGSS operates as a protection mechanism for all the Datacryptor 2000's sensitive contents (keys, sensitive algorithms etc.), by providing resistance to physical intrusion and voltage attacks, and temperature and motion sensors.

Intrusion protection is provided by a copper mesh that consists of two circuits - a continuity circuit and a guard circuit. The SGSS is surrounded by the mesh and potted in an opaque resin. An alarm is triggered if the continuity circuit is broken or if the two circuits are bridged. The wires of the circuit are lacquered so that they cannot bridge simply by touching. Any attempt to drill through the resin and mesh should result in an alarm being triggered either by breaking the continuity circuit or by shorting

the two circuits. Similarly, attempting to dissolve the resin to gain access to the secure area would dissolve the lacquer on the wires of the mesh, causing the two circuits to short.

The alarm circuit is powered from the main power supply when this is available, or by a battery otherwise. Should the battery fail or become disconnected (i.e. voltage drops), an alarm will be triggered. Similarly, if the voltage levels surge or are actively driven above the normal levels, an alarm is triggered. This prevents both high and low voltage attacks.

A temperature sensor causes the alarm circuit to be triggered at temperatures above 60°C or below -5°C, and a movement sensor triggers an alarm on detection of movement. In the evaluated configuration the temperature sensor is enabled. (Note that the motion sensor is unlikely to respond to a small movement of the unit.)

The effect of triggering an alarm is to force the voltage supply rails to all devices containing sensitive information to ground, causing them to lose their contents. Additionally the interface lines into the DC2K specific hardware are disconnected to prevent an attacker from attempting to prevent device erasure by externally driving in supply voltage.

Inter-document Reference – SF_SGSS_alarm_circuitry

9.1.4 DC2K Application

The DC2K application is responsible for several security critical functions, discussed below:

- cryptographic authentication of key exchange algorithm
- cryptographic authentication of encryption algorithm
- cryptographic authentication of Certificate Authorities
- cryptographic authentication of Key Exchange Algorithm Keysets

Note that all four functions employ the same data authentication implementation (managed in one library made available to the SGSS application and the DC2K application) as described in section 9.1.2.

9.1.4.1 Authentication of Key Exchange Algorithm

The DC2K application contains an implementation of a data authentication algorithm as listed in section 4.1.2. In addition, at manufacture time, the "DC2K application" public key value is embedded in the DC2K application. When a Key Exchange Algorithm is generated, a digital signature is generated over the algorithm using the corresponding secret key, which is held securely at the development site. The algorithm is concatenated with the signature.

On loading the Key Exchange Algorithm, the signature concatenated with the algorithm is verified by the DC2K application's data authentication implementation using the public key embedded within it. If the verification is successful, the algorithm is loaded into the SGSS hardware and may be used. Otherwise, the algorithm is rejected and cannot be loaded into the unit.

In this way, only Key Exchange Algorithms that have been signed by an authorised body may be run in the unit.

Inter-document Reference SF_DC2K_data_authentication_implementation

9.1.4.2 Authentication of Encryption Algorithm

The DC2K application contains an implementation of a data authentication algorithm as listed in section 4.1.2. In addition, at manufacture time, the "DC2K application" public key value is embedded in the DC2K application. When an Encryption Algorithm is generated, a digital signature is generated over the algorithm using the corresponding secret key, which is held securely at the development site. The algorithm is concatenated with the signature.

On loading the Encryption Algorithm, the signature concatenated with the algorithm is verified by the DC2K application's data authentication implementation using the public key embedded within it. If the verification is successful, the algorithm is loaded into the SGSS hardware and may be used. Otherwise, the algorithm is rejected and cannot be loaded into the unit.

In this way, only Encryption Algorithms that have been signed by an authorised body may be run in the unit.

Inter-document Reference SF_DC2K_data_authentication_implementation

9.1.4.3 Authentication of Certificate Authorities

Authentication of Certificate Authorities occurs when signed Certificate Authorities are loaded during the unit's commissioning process or at a later stage. The signature on the Certificate Authority is verified by the DC2K's data authentication implementation, and may only be loaded and subsequently used if the validation is successful. If the CA is loaded, its public key may be used in turn to verify the signatures on algorithms and key exchange keysets.

Inter-document Reference SF_DC2K_data_authentication_implementation

9.1.4.4 Authentication of Key Exchange Algorithm Keysets

Authentication of Key Exchange Algorithm Keysets occurs at two points in Datacryptor 2000 operation:

9.1.4.4.1 Unit Commissioning

During the unit's commissioning process, Key Exchange Algorithm Keysets signed by a CA are loaded. The signature on the keyset is verified by the DC2K's data authentication implementation, and may only be loaded and subsequently used if the validation is successful.

9.1.4.4.2 Key Exchange Protocol

During the key exchange protocol (see [2], section 4.2.1), units exchange signed key exchange certificates. Both units must positively verify that the keyset has been authorised by a CA that they are operating under before proceeding to generate a shared key encryption key. The DC2K application's data authentication implementation is used for this purpose

Note that this procedure is identical whether a unit is communicating with another unit, or with the management centre.

Inter-document Reference SF_DC2K_data_authentication_implementation

9.1.5 DC2K Key Exchange Algorithm

A secure key exchange algorithm allows two units, (or a unit and a management centre) to establish a common Key Encryption Key (KEK) without either party having to transmit any secret data.

An implementation of a secure key exchange algorithm, as listed in section 4.1.2, is used for this purpose, which requires the input of both parties' signed public and secret keys. In addition to these values, each unit inputs a random one-time public-secret key pair (using the SGSS random number generator), ensuring that every KEK generated between the two parties is unique.

This algorithm is discussed in detail in [2], sections 4.2.2 (steps 1 – 8) and 8.2.

Inter-document Reference SF_DC2K_key_exchange_algorithm

9.1.6 DC2K Encryption Algorithm

The Datacryptor 2000 uses an encryption algorithm for two purposes – key encryption and data encryption for user and management traffic.

9.1.6.1 Key Encryption

Having agreed a KEK as described in section 9.1.5, the two units (or a unit and its management centre) must securely derive a data encryption key (DEK). This is achieved by both entities generating random data, encrypting it with the KEK, and sending it to the other party.

The encrypted random data is decrypted by both entities, and combined to generate a shared DEK.

This is described in section 4.1.1 (steps 1 – 5) of [2].

9.1.6.2 Data Encryption

Having agreed a DEK, the encryption algorithm may now be used to encrypt transmitted user (or management) data and decrypt received user (or management) data.

Inter-document Reference SF_DC2K_encryption_algorithm

9.1.7 Unit Management

A unit may be managed (i.e. its security and communications attributes changed) by use of the management centre previously discussed. The DC2K and the management centre communicate in the same way that two units communicate; firstly a cryptographic key must be established according to the

key management protocol described in [2], and subsequently, all traffic sent between the management centre and the unit is subject to encryption.

To provide this functionality, the management centre has the encryption and key exchange capabilities that are equivalent to those of the Datacryptor 2000 unit. However, the unit (rather than the management centre) enforces security by failing to act on a management request if it cannot decrypt it.

In order to be able to successfully decrypt data sent to it by the management centre, both entities must be using the same key for data encryption. Similarly, in order to agree a common key, both entities must successfully complete the key exchange protocol. This requires the management centre to have access to key material that has been authorised by the same CA as the unit is operating under.

It is assumed that only authorised individuals have access to such key material.

9.2 Assurance Measures

In the sections that follow, non-italic font is used to state the developer actions of the assurance requirements (extracted directly from [1]), and *italic font* describes the evaluation deliverables that will provide the necessary assurance.

9.2.1 ACM_AUT.1 Partial CM automation

Developer action elements
ACM_AUT.1.1D The developer shall use a CM system.
ACM_AUT.1.2D The developer shall provide a CM plan.

TOE development conforms to the Project Filing Procedure provided at [3]. The plan describes the use of automated tools for configuration management.

9.2.2 ACM_CAP.4 Generation support and acceptance procedures

Developer action elements
ACM_CAP.4.1D The developer shall provide a reference for the TOE.

A unique TOE reference is provided in section 4.1.

ACM_CAP.4.2D The developer shall use a CM system.
ACM_CAP.4.3D The developer shall provide CM documentation.

TOE development conforms to the Project Filing Procedure at [3]. The procedure describes the use of automated tools for configuration management. References [4] and [5] provide a Configuration List and the procedure for Managing Projects in the Engineering Group respectively. Documentation describing the use of an automated version control system is provided at [6] Problem Reporting and Change Control.

9.2.3 ACM_SCP.2 Development tools CM coverage

Developer action elements
ACM_SCP.2.1D The developer shall provide CM documentation.

Documentation management conforms to the Project Filing Procedure described at [3]. In addition, references [7] and [8] discuss the production of documentation and tracking of software tools respectively.

9.2.4 ADO_DEL.2 Detection of modification

Developer action elements
ADO_DEL.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.
ADO_DEL.2.2D The developer shall use the delivery procedures.

TOE delivery conforms to the company Packing and Despatch procedures provided at reference [9].

9.2.5 ADO_IGS.1 Installation, generation, and start-up procedures

Developer action elements
ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

The TOE is shipped with a manual. This is provided at reference [10] and [11].

9.2.6 ADV_FSP.2 Fully Defined External Interfaces

Developer action elements
ADV_FSP.2.1D The developer shall provide a functional specification.

A functional specification of the TOE Security Functions is provided at [28].

9.2.7 ADV_HLD.2 Security enforcing high-level design

Developer action elements
ADV_HLD.2.1D The developer shall provide the high-level design of the TSF.

A high-level design of the TOE Security Functions is provided at [25]. .

9.2.8 ADV_IMP.1 Subset of the Implementation of the TSF

Developer action elements
ADV_IMP.1.1D The developer shall provide the implementation representation for a selected subset of the TSF.

The TSF implementation representation shall be provided in the form of C source, VHDL (Very High Speed Integrated Circuit Hardware Description Language), and hardware schematics for the version under evaluation.

9.2.9 ADV_LLD.1 Descriptive low-level design

Developer action elements
ADV_LLD.1.1D The developer shall provide the low-level design of the TSF.

A descriptive low-level design of the TOE Security Functions is provided at [24].

9.2.10 ADV_RCR.1 Informal correspondence demonstration

Developer action elements
ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

Reference [14] provides an analysis of the correspondence between all adjacent pairs of the TSF.

9.2.11 ADV_SPM.1 Informal TOE security policy model

Developer action elements
ADV_SPM.1.1D The developer shall provide a TSP model.
ADV_SPM.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model.

A TOE Security Policy Model, along with a correspondence with the TOE functional specification, is provided in [15].

9.2.12 AGD_ADM.1Administrator guidance

Developer action elements
AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

The TOE is shipped with a manual. This is provided at reference [10] and [11].

9.2.13 AGD_USR.1 User guidance

Developer action elements
AGD_USR.1.1D The developer shall provide user guidance.

The TOE is shipped with a manual. This is provided at reference[10] and [11]. (Note, however, that users will have no direct interaction with the TOE – see 6.1.3.4. This means that in the context of this evaluation there is no user documentation.)

9.2.14 ALC_DVS.1 Identification of security measures

Developer action elements
ALC_DVS.1.1D The developer shall produce development security documentation.

Development of the TOE conforms to the standards defined within the UK Government's Manual of Protective Security.

9.2.15 ALC_LCD.1 Developer defined life-cycle model

Developer action elements
ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.
ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

A life-cycle model for the Datacryptor 2000 is defined at [5]..

9.2.16 ALC_TAT.1 Well-defined development tools

Developer action elements
ALC_TAT.1.1D The developer shall identify the development tools being used for the TOE.
ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of the development tools.

TOE development tools used are described in the software tools registry, as defined in [8]. The coding standards used are described at references [17] and [18] ('C' and VHDL coding standards respectively)

9.2.17 ATE_COV.2 Analysis of coverage

Developer action elements
ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

An analysis of the Datacryptor 2000 Test Coverage is provided at reference [19].

9.2.18 ATE_DPT.1 Testing: high-level design

Developer action elements
ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

An analysis of the depth of testing carried out on the TOE is provided at reference [20]. The analysis includes a description of the correspondence between the tests performed and the high level design specification.

9.2.19 ATE_FUN.1 Functional testing

Developer action elements
ATE_FUN.1.1D The developer shall test the TSF and document the results.
ATE_FUN.1.2D The developer shall provide test documentation.

The Datacryptor 2000 has been tested as defined in references [21] and [22] inclusive and [29], which include test results.

9.2.20 ATE_IND.2 Independent testing - sample

Developer action elements
ATE_IND.2.1D The developer shall provide the TOE for testing.

The TOE is available for independent testing as required.

9.2.21 AVA_MSU.2 Validation of analysis

Developer action elements
AVA_MSU.2.1D The developer shall provide guidance documentation**.**
AVA_MSU.2.2D The developer shall document an analysis of the guidance documentation.

Guidance documentation is provided at reference [10] and [11]. An analysis of the guidance documentation is given at reference [23].

9.2.22 AVA_SOF.1 Strength of TOE security function evaluation

Developer action elements
AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

All TOE Security Functions realised by a probabilistic or permutational mechanism are cryptographic. Hence a statement regarding their strength level is outside the scope of this Security Target.

9.2.23 AVA_VLA.2 Developer vulnerability analysis

Developer action elements
AVA_VLA.2.1D The developer shall perform and document an analysis of the TOE deliverables searching for ways in which a user can violate the TSP.
AVA_VLA.2.2D The developer shall document the disposition of identified vulnerabilities.

A Vulnerability Analysis has been carried out on the Datacryptor 2000. This is provided at reference [27].

10 Protection Profile Claims

No claims of conformance to a Protection Profile are made.

11 Rationale

11.1 General Statement

The security objectives are designed to counter the threats in line with the threat descriptions given in section 6.2.

11.2 Security Objectives Rationale

Threat	Asset	TOE Security Objective	Environment Security Objective
T_extract_data_from_ secure_domain	*A_user_data*		*OBE_protect_secure_ domain*
T_record_plaintext_data _from_insecure_domain	*A_user_data*		*OBE_transmit_data_ through_TOE*
			OBE_apply_suitable_ TOE_mode_to_data
T_cryptanalyse_data_ within_insecure_domain	*A_user_data*	*OBT_DC2K_provide_ data_confidentiality*	
T_access_to_secret_ authentication_key	*A_user_key*		*OBE_protect_key_ material*
	A_user_data		
T_access_to_secret_key _exchange_alg_keys	*A_user_key*		*OBE_protect_key_ material*
	A_user_data		
T_access_to_keys_ within_unit	*A_user_key*	*OBT_SGSS_provide_ resistance_to_ physical_attack*	
	A_user_data		
T_access_to_algorithm_ within_unit	*A_user_algorithm*	*OBT_SGSS_provide_ resistance_to_ physical_attack*	
T_access_to_algorithm_ outside_unit	*A_user_algorithm*		*OBE_protect_ algorithms*
T_cryptanalyse_keys_ within_insecure_domain	*A_user_key*	*OBT_DC2K_provide_ secure_key_ management*	
	A_user_data		
T_loss_of_ commissioned_unit	*A_user_data*		*OBE_protect_keyed_ unit*
T_tamper_with_unit	*A_user_data*		*OBE_check_for_unit _tamper* *OBE_protect_keyed_ unit*
*T_application_*	*A_user_algorithm*	*OBT_SGSS_provide_*	

replacement	*A_user_key* *A_user_data*	*secure_application_ load*	
T_algorithm_ replacement	*A_user_key* *A_user_data*	*OBT_DC2K_provide_ secure_algorithm_ load*	
T_certificate_authority_ replacement	*A_user_key* *A_user_data*	*OBT_DC2K_provide_ secure_CA_load*	
T_key_exchange_ certificate_replacement	*A_user_key* *A_user_data*	*OBT_DC2K_provide_ secure_key_exchange _keyset_load*	

Table 1 – correlation between threats and security objectives required to fully counteract threat

Table 1 lists each of the threats identified in section 6.2, and for each, applies sufficient security objectives (taken from section 7) to fully counteract the threat. Since every threat is counteracted, the security objectives are sufficient to meet all of the assumed threats.

The following shows that, for each threat, the security objectives applied to it in Table 1 successfully counter that threat.

The threat that an attacker with physical access to the secure domain gains access to data assets residing there (*T_extract_data_from_secure_domain*) is countered by *OBE_protect_secure_domain*, which applies physical protection measures to the secure domain.

The threat that an attacker is able to access sensitive data because it has been sent unencrypted into the insecure domain (*T_record_plaintext_data_from_insecure_domain*) is countered by two security objectives, *OBE_transmit_data_through_TOE* and *OBE_apply_suitable_TOE_mode_to_data.* The first ensures that all data transmitted from the secure domain to the insecure domain passes through the TOE. The second ensures that the TOE encrypts sensitive data passing through it. Clearly, if both objectives are met then the threat is removed because sensitive data will not exist in an unencrypted form within the insecure domain.

The threat that an attacker gains access to data by recording it in its encrypted form while in the insecure domain and then employing cryptanalysis (*T_cryptanalyse_data_within_insecure_domain*) is countered by the security objective *OBT_DC2K_provide_data_confidentiality*, which implements a confidentiality service by the use of encryption. If correctly implemented, this objective diminishes the threat by requiring greater expertise and resources on the part of the attacker.

The threat of disclosure of the externally held secret authentication key (*T_access_to_secret_authentication_key*) makes a man-in-the-middle attack possible. This threat is countered by the security objective *OBE_protect_key_material*, which ensures the appropriate physical protection measures are used for key material stored externally to the TOE.

The threat of disclosure of secret key exchange algorithm keys that are loaded into the TOE from an external source (*T_access_to_secret_key_exchange_alg_keys*) is also countered by *OBE_protect_key_material*, which applies physical security measures to key material stored externally.

The threat of an attacker gaining knowledge of secret key material stored within the TOE (*T_access_to_keys_within_unit*) is countered by *OBT_SGSS_provide_resistance_to_physical_attack*, which provides resistance to such forms of direct physical attack.

Similarly, the threat of an attacker gaining knowledge of sensitive algorithms while they are stored within the TOE (*T_access_to_algorithm_within_unit*) is also countered by *OBT_SGSS_provide_resistance_to_physical_attack* as it provides resistance to such forms of direct physical attack

The threat of an attacker gaining knowledge of sensitive algorithms while they are external to the TOE (*T_access_to_algorithm_outside_unit*) is countered by the objective *OBE_protect_algorithms*, which ensures that appropriate physical security measures are applied to algorithms stored externally to the TOE.

The threat of an attacker determining key encryption keys or data encryption keys by cryptanalysis of the key exchange protocol between two instances of the TOE (*T_cryptanalyse_keys_within_insecure_domain*) is countered by security objective *OBT_DC2K_provide_secure_key_management*, which provides a means of exchanging key material securely. If this objective is met correctly then the threat is greatly diminished because the method of attack becomes impractical.

The threat of an attacker gaining possession of a commissioned unit (*T_loss_of_commissioned_unit*) is countered by security objective *OBE_protect_keyed_unit*, which applies physical protection measures to commissioned units. This reduces the opportunity and so diminishes the threat.

The threat of an attacker physically tampering with the TOE so that it did not encrypt transmitted data (*T_tamper_with_unit*) is countered by two, security objectives *OBE_check_for_unit_tamper* and *OBE_protect_keyed_uni*t.

The first ensures that units are checked periodically for signs of tampering. The objective will mitigate the effects of the threat by ensuring that such an attack is detected. The second applies physical protection measures to commissioned units. This reduces the opportunity and so diminishes the threat.

The threat of an attacker subverting the security of the TOE by installing their own application (*T_application_replacement*) is countered by the TOE security objective *OBT_SGSS_provide_secure_application_load*, which provides a means of cryptographically verifying the authenticity of an application. Implemented correctly, this objective greatly reduces the likelihood of this attack being successful.

Similarly, the threat of an attacker installing a rogue encryption or key exchange algorithm (*T_algorithm_replacement*) is countered by the TOE security objective *OBT_DC2K_provide_secure_algorithm_load*, which provides a means of cryptographically verifying the authenticity of an algorithm prior to its loading and use. This reduces the likelihood of a successful attack.

If an attacker manages to load their own Certificate Authorities into two communicating instances of the TOE, it may lead to the exposure of key material and ultimately to the exposure of user data. This threat (*T_certificate_authority_replacement*) is countered by the TOE security objective

OBT_DC2K_provide_secure_CA_load, which provides a means of cryptographically verifying the authenticity of a CA before its loading and use. This reduces the likelihood of a successful attack.

Likewise, the threat of an attacker loading known Key Exchange certificates so that keys and user data is exposed (*T_key_exchange_certificate_replacement*) is countered by the TOE security objective *OBT_DC2K_provide_secure_key_exchange_keyset_load*, which provides a means of cryptographically verifying the authenticity of a Key Exchange Keyset prior to its loading and use. This reduces the likelihood of a successful attack.

Therefore, the security objectives are suitable to counter all identified threats.

11.3 Security Requirements Rationale

11.3.1 Environment Assumptions

Environment Objectives	Environment Assumptions defined in section 6.1
OBE_protect_secure_domain	*6.1.1.1, 6.1.2*
OBE_transmit_data_through_TOE	*6.1.1.1, 6.1.3.3*
OBE_apply_suitable_TOE_mode_to_data	*6.1.1.1, 6.1.3.1*
OBE_protect_key_material	*6.1.3.2, 6.1.3.4*
OBE_protect_algorithms	*6.1.3.2, 6.1.3.4*
OBE_protect_keyed_unit	*6.1.3.2, 6.1.3.4*
OBE_check_for_unit_tamper	*6.1.1*

Table 2 – Environment Assumptions required to meet each Environment Objective

Table 2 lists each of the environment objectives identified in section 7, and for each, applies sufficient environment assumptions (taken from section 6.1) to meet the objective. Since every environment objective is met, the environment assumptions are sufficient to meet all of the environment objectives.

In addition, where each objective is fulfilled by exactly one assumption, it follows that the assumptions must be necessary as well as sufficient to meet the objective.

While some objectives require more than one assumption to be fully met, it is clear that these are consistent with the standard operation of a secure environment. For example, protection of key material, as identified by objective *OBE_protect_key_material* requires *both*

- procedural measures to physically protect key material (6.1.3.2), and
- trusted personnel to perform this duty (6.1.3.4),

and the same arguments follow for objectives *OBE_protect_algorithms* and *OBE_protect_keyed_unit*.

The objective *OBE_check_for_unit_tamper* requires that:

- action be taken in the event of suspected tampering (6.1.1).

The objective *OBE_protect_secure_domain* requires both that:

- the secure environment is protected by appropriate means (6.1.1.1), and that
- the value of the assets protected by the TOE should be appropriate for the EAL4 assurance level claimed for it: this will be relatively high, and can only be determined by the administrator in light of local conditions (6.1.2).

The objective *OBE_transmit_data_through_TOE* requires both that:

- sensitive data is transmitted through the TOE (6.1.1.1), and that
- suitable connectivity exists to allow this to occur (6.1.3.3).

Finally, the application of a suitable TOE mode to sensitive data, as defined by objective *OBE_apply_suitable_TOE_mode_to_data* requires that:

- a suitable mode to be applied to sensitive data (6.1.1.1), and that
- a secure management capability by which this action may be taken (6.1.3.1)

These arguments demonstrate the consistency of the environment assumptions with respect to the environment objectives.

11.3.2 Functional Requirements

Security Objective	IT Functional Requirement
OBT_DC2K_provide_data_confidentiality	FCS_COP.1
OBT_DC2K_provide_secure_ key_management	FCS_CKM.1, FCS_CKM.2, FCS_COP.1
OBT_SGSS_provide_resistance_to_physical_a ttack	FPT_PHP.3
OBT_SGSS_provide_secure_application_ load	FCS_COP.1
OBT_DC2K_provide_secure_algorithm_ load	FCS_COP.1
OBT_DC2K_provide_secure_CA_load	FCS_COP.1
OBT_DC2K_provide_secure_key_ exchange_keyset_load	FCS_COP.1

Table 3 - IT functional requirements required to meet each of the TOE's security objective.

Table 3 lists each of the security objectives identified in section 7, and for each, applies sufficient IT functional requirements (taken from section 8.1.1) to meet the objective. Since every security objective is met, the IT functional requirements are sufficient to meet all of the security objectives.

In addition, where each objective is met by exactly one IT functional requirement, it follows that the functional requirement must be necessary as well as sufficient to meet the objective.

Each security objective from Table 3 is considered below where the indicated IT functional requirements are shown to meet it.

The objective *OBT_DC2K_provide_data_confidentiality* is for the TOE to provide the option of a confidentiality service to all data transmitted through it. Data confidentiality is achieved using encryption and is fully covered by the data encryption component of the functional requirement FCS_COP.1 (Cryptographic Operation).

The objective *OBT_DC2K_provide_secure_key_management* requires three IT functional requirements to fully meet it, FCS_CKM.1 (Cryptographic Key Generation), FCS_CKM.2 (Cryptographic Key Distribution) and FCS_COP.1 (Cryptographic Operation). Public data exchange is achieved by FCS_COP.1 (see [2] section 4.2.1). FCS_CKM.1 and FCS_CKM.2 respectively achieve generation and distribution of keys. Logically, all three requirements are needed to satisfy this objective fully.

The objective *OBT_SGSS_provide_resistance_to_physical_attack* is for the TOE to provide resistance to direct physical attacks aimed at extracting sensitive data. The functional requirement FPT_PHP.3 (Resistance to Physical Attack) satisfies this objective by resisting the physical tampering scenarios listed in its definition.

The remaining four objectives (*OBT_SGSS_provide_secure_application_load, OBT_DC2K_provide_secure_algorithm_load, OBT_DC2K_provide_secure_CA_load, OBT_DC2K_provide_secure_key_exchange_keyset_load*) are all concerned with providing a means to cryptographically verify the authenticity of a piece of data (namely a Datacryptor 2000 application, a cryptographic algorithm, a Certificate Authority, and a Key Exchange Keyset). All are satisfied by the data authentication component of the functional requirement FCS_COP.1 (Cryptographic Operation).

11.3.3 Dependencies of Functional Requirements

Reference [1] states that some IT functional requirements are dependent on others (and in addition, some dependencies themselves have further dependencies), as shown below:

IT functional Requirements	Dependencies
FCS_CKM.1	FCS_CKM.2
	FCS_CKM.4
	FMT_MSA.2
FCS_CKM.2	FCS_CKM.1
	FCS_CKM.4
	FMT_MSA.2

FCS_CKM.4	FCS_CKM.1
	FMT_MSA.2
FMT_MSA.2	ADV_SPM.1
	FDP_ACC.1
	FMT_MSA.1
	FMT_SMR.1
(ADV_SPM.1)	(see section 11.3.4)
FDP_ACC.1	FDP_ACF.1
FMT_SMR.1	FIA_UID.1
FMT_MSA.1	FDP_ACC.1
	FMT_SMR.1
FDP_ACF.1	FMT_MSA.3
	FDP_ACC.1
FIA_UID.1	No dependency
FMT_MSA.3	FMT_MSA.1
	FMT_SMR.1
FCS_COP.1	FCS_CKM.1
	FCS_CKM.4
	FMT_MSA.2
FPT_PHP.3	No dependency

Table 4 – IT Functional Requirements Dependencies, with claimed IT functional requirements in **bold type**.

Table 4 shows the dependencies of components (with iterated dependencies of dependencies). It should be noted that only those components in bold, i.e. FCS_CKM.1, FCS_CKM.2, FCS_COP.1 and FPT_PHP.3 have been claimed as IT functional requirements for the Datacryptor 2000. Taking each (unclaimed) dependency in turn, the following sections provide a rationale as to why these dependencies are inappropriate and/or irrelevant in the context of the Datacryptor 2000 evaluation.

11.3.3.1 Cryptographic Key Destruction

FCS_CKM.4

The Datacryptor 2000 disables cryptographic keys as a result of key expiry or a deletion request from the user. However, such disabling does not constitute “key destruction” as such, it simply ensures that the keys are unavailable for subsequent use by the product.

Unlike a standard software system or product, the Datacryptor employs physical protection measures to prevent both unauthorised and authorised access to cryptographic key values (see section 8.1.1.4). This means that FCS_CKM.4 is effectively subsumed by FPT_PHP.3 (Resistance to Physical Attack). Thus dependency FCS_CKM.4 is not relevant in the context of Datacryptor 2000.

11.3.3.2 Management of Security Attributes

FMT_MSA.2

In the manner in which they are described in [1], management of security attributes, is a function that is most meaningful in the context of a typical software security product or system. In such a situation, users might log on to administratively assigned accounts using unique user IDs and passwords, and they may be restricted to only performing certain actions (e.g. read, write) on files with certain ownership criteria (e.g. owner, group, all).

In the context of the TOE however the capability to view and alter security attributes such as key lifetimes, unit alarm settings etc., (rather than to the information under protection itself) is performed by establishing an encrypted "management session" between a management centre and the unit under management. Individuals performing these tasks are simply considered as authorised or unauthorised, and as stated in section 9.1.7, it is assumed that only authorised individuals have access to the key material. Individuals without access to the appropriate key material (i.e. unauthorised individuals) are unable to manage the box in such a way as to view or alter sensitive information.

In this way, the Datacryptor's claimed IT functional requirements of cryptographic operation, cryptographic key distribution, cryptographic key generation and resistance to physical attack provide all the necessary support for the dependency FMT_MSA.2. In the manner in which it is described in [1], this dependency is inappropriate for this target of evaluation.

11.3.4 Assurance Requirements

The assurance requirements specified in this security target are exactly those specified by the Evaluation Assurance Level 4. The actual Evaluation Assurance Level required for the evaluation is specified in section 8.1.2 It may be any EAL up to and including 4. Where the EAL is 4, this set is consistent and mutually supportive. All dependencies are implicitly met by inclusion of the dependent component itself or a stronger component from the same assurance family within the set. Where the EAL is lower than 4, the actual evaluation assurance requirements are a subset of those at EAL 4, and are therefore at least met, if not exceeded by those specified in this document.

Choice of the assurance component set of the EAL defined in section 8.1.2 is appropriate to the evaluation for company marketing reasons.

11.3.5 Security Requirements are Mutually Supportive and Internally Consistent

The security requirements do not conflict as they apply to distinct but related operations. FCS_CKM.1 and FCS_CKM.2 apply to the generation and distribution of keys respectively. They support each other in the overall objective of secure key management.

FCS_COP.1 applies to the operations of data authentication, data encryption, and key exchange protocol. These do not conflict with FCS_CKM.1 and FCS_CKM.2.

The final security requirement, FPT_PHP.3, is concerned with resistance to physical attack and clearly does not conflict with the other requirements, all of which are related to cryptographic services.

The preceding shows that the set of security requirements is internally consistent. It also shows that they are mutually supportive in that they support each other where necessary.

11.4 Target of Evaluation Summary Specification Rationale

11.4.1 Satisfaction of TOE Security Functional Requirements

IT Functional Requirements	TOE Security Functions
FCS_CKM.1	*SF_SGSS_Random_Number_Generator* *SF_DC2K_key_exchange_algorithm* *SF_DC2K_encryption_algorithm*
FCS_CKM.2	*SF_DC2K_key_exchange_algorithm*
FCS_COP.1	*SF_SGSS_data_authentication_implementation*
	SF_DC2K_data_authentication_implementation
	SF_DC2K_encryption_algorithm
	SF_DC2K_key_exchange_algorithm
FPT_PHP.3	*SF_SGSS_alarm_circuitry*

Table 5 – Use of TOE Security Functions to meet IT functional Requirements

Table 5 lists each of the IT Functional Requirements identified in section 8.1.1, and identifies all the TOE security functions needed to meet that requirement. Since every requirement is met by one or more security functions, the security functions are sufficient to meet all of the IT Functional Requirements.

In addition, where each IT Functional Requirement is met by exactly one security function (and assuming the requirement is valid), it follows that the security functions must be necessary as well as sufficient to counter the threat.

The suitability of the security functions to meet the IT Functional Requirements is shown as follows.

The functional requirement FCS_CKM.1 (Cryptographic Key Generation) is met by the security functions *SF_SGSS_Random_Number_Generator*, *SF_DC2K_key_exchange_algorithm* and *SF_DC2K_encryption_algorithm*. *SF_SGSS_Random_Number_Generator* provides a hardware random number generator for use by *SF_DC2K_key_exchange_algorithm* and *SF_DC2K_encryption_algorithm* in the generation of KEKs and DEKs respectively.

The functional requirement FCS_CKM.2 (Cryptographic Key Distribution) is met by the security function *SF_DC2K_key_exchange_algorithm* which provides an implementation of a secure key exchange algorithm (see 9.1.5).

The functional requirement FCS_COP.1 (Cryptographic Operation) is met by four security functions, namely *SF_DC2K_key_exchange_algorithm*, *SF_DC2K_encryption_algorithm*, *SF_DC2K_data_authentication_implementation*, and

SF_SGSS_data_authentication_implementation. It is clear from previous sections of this document (9.1 and 11.3 for example), that all four functions are required to support the security objectives met by FCS_COP.1. Therefore the four security functions are necessary and sufficient to counter the threat.

Finally the functional requirement FPT_PHP.3 (Resistance to Physical Attack) is met by the security function *SF_SGSS_alarm_circuitry* as can be seen by comparing 8.1.1.4 and 9.1.3.2.

It follows that the set of security functions is both necessary and sufficient to support the IT Functional Requirements. Note also that the security functions work together so as to satisfy the Functional Requirements (i.e. the security functions do not conflict with each other and are mutually supportive in satisfying the Functional Requirements).

11.4.2 Compliance of Assurance Measures with Assurance Requirements

Section 9.2 lists every assurance requirement, and separately addresses the assurance measures for each. Since this provides a one-to-one mapping between assurance requirements and assurance measures, (and assuming the suitability of each assurance measure), the set of assurance measures must provide full compliance with the set of assurance requirements.